Programming from an Empirical Modelling perspective 2

From modelling with definitive scripts to programming:

- representing state in programming
- behaviour of programs
- the semantics of programs

State

States relevant to programming ...

- state within the executing program
- external state: what is visible?
- state in respect of interaction
- state in program development
- state significant in the external world

- Diverse representations are required:
- *state within the executing program*
  - Program variables, machine locations
- *- external state: what is visible?*
  - Graphical techniques
- *- state in respect of interaction*
  - Statechart, message sequence diagram

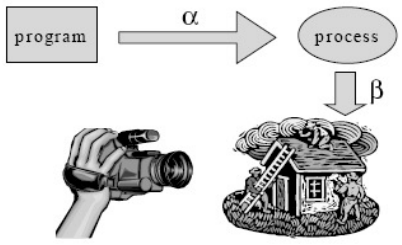
- Diverse representations required …
- *state in program development*

  UML diagrams, prototypes
- *state significant in the external world*

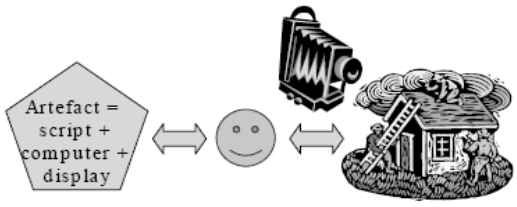
  apprehended by the human interpreter

cf. Brian Cantwell-Smith on semantics …

Semantic Relations (I)



Semantic Relations (II)



States within oxoGardner1999

Definitive scripts express …
- internal state – contents of squares
- visible state – appearance of the board
- interaction state: whose turn is it?
- state of development
- state of mind of the player: which square?

(demo)

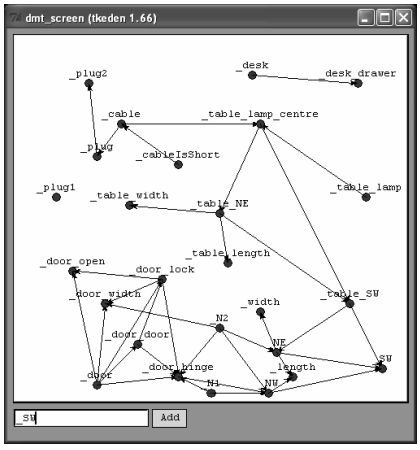
Modelling with definitive scripts:
… a holistic view of state that integrates and conflates all the different perspectives

in contrast to

Programming-in-the-wild:
… an eclectic model of state in which many different strategies for representation and interpretation are jumbled up together

Objects and dependencies

- An **object** corresponds to a particular way of associating observables: grouping together observables according to whether they exist concurrently
- A **dependency** links observables according to how they are linked in change: whether making a change to the value of one observable necessarily entails changing others



Object model vs. account of observation

An account of observation is a more primitive concept than an object model: it entails fewer preconceptions about what might be observed …

"**Definitive scripts are neutral wrt agent's views & privileges**"

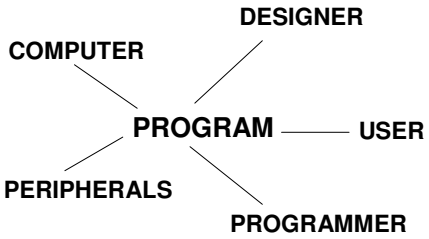
Object model vs. account of observation 2

Definitive script expresses different agent views and privileges to transform

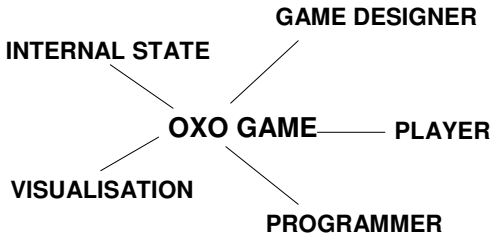
(cf. subject-oriented programming)

"What architect can do vs what user can do"

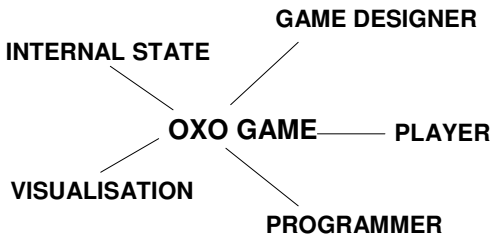
… highlights how the script affords *views of* and *access to* possible transformations



… compare this with the OXO laboratory

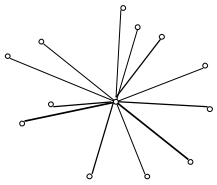


… Behaviour as programmed state change



Static and dynamic elements of state

Definitive scripts as “furry blobs”



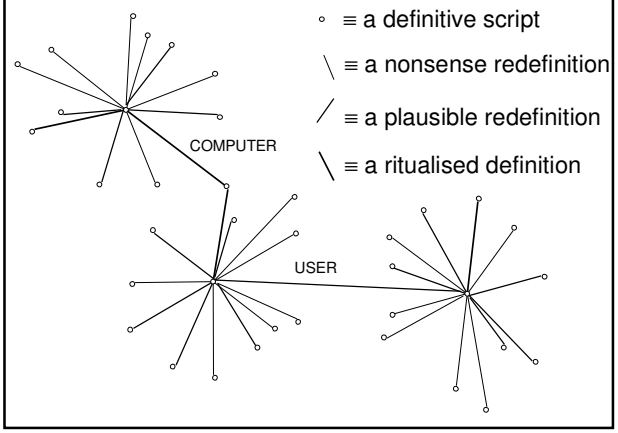
◦ ≡ a definitive script

\ ≡ a nonsense redefinition

/ ≡ a plausible redefinition

\ ≡ a ritualised definition

Plausible : *could* open the desk drawer
– note continuous spectrum of redefinitions
Ritualised : door *automatically* closes after being opened
Nonsense : opening the drawer makes the room smaller



Classical programming …1

Behaviour is derived from a pre-specified conception of function and purpose …

… based on interactions whose outcomes are reliable and for which the mode of interpretation is determined in advance

…motivates declarative approaches

Classical programming …2

… motivates declarative approaches:

```
output=F(input)
```

… problematic to deal with a dynamic input, as in playing a game

… hence add “lazy evaluation” to model as

```
stream_of_output=F(stream_of_input)
```